

SQL skills used in this project to aggregate the data as required. Specific used functions are-  
EXTRACT(MONTH FROM order\_date) for month. GROUP BY, SUM() for revenue, COUNT(DISTINCT  
order\_id) for volume, ORDER BY for sorting, and then Limit results for specific time periods.

## Queries Output-

--Count of rows

Select count(\*) As Total\_Rows from Data;

The screenshot shows a SQL query editor with the following code:

```
1 use [ Sales_Trend];
2 select * from Data;
3
4
5 --Data Exploration
6
7 --Count of rows
8 Select count(*) As Total_Rows from Data; --240
9
10 --Sample data
11 Select TOP 10 * FROM Data ORDER BY Total_Revenue;
12
13 --Null values
14 Select * FROM Data WHERE Transaction_ID IS NULL OR Date IS NULL OR Product_Category IS NULL OR Product_Name IS NULL
15 OR Units_Sold IS NULL OR Unit_Price IS NULL OR Total_Revenue IS NULL OR Region IS NULL OR Payment_Method IS NULL;
16
17 --No nulls present in the dataset
18
19
```

The results pane shows a single row with the value 240 for Total\_Rows.

Total_Rows
240

Select TOP 10 \* FROM Data ORDER BY Total\_Revenue;

The screenshot shows the same SQL query editor with the following code:

```
1 use [ Sales_Trend];
2 select * from Data;
3
4
5 --Data Exploration
6
7 --Count of rows
8 Select count(*) As Total_Rows from Data; --240
9
10 --Sample data
11 Select TOP 10 * FROM Data ORDER BY Total_Revenue;
12
13 --Null values
14 Select * FROM Data WHERE Transaction_ID IS NULL OR Date IS NULL OR Product_Category IS NULL OR Product_Name IS NULL
15 OR Units_Sold IS NULL OR Unit_Price IS NULL OR Total_Revenue IS NULL OR Region IS NULL OR Payment_Method IS NULL;
16
17 --No nulls present in the dataset
18
19
```

The results pane shows the top 10 rows of the Data table, ordered by Total\_Revenue.

Transaction_ID	Date	Product_Category	Product_Name	Units_Sold	Unit_Price	Total_Revenue	Region	Payment_Method	
1	10107	2024-04-16	Beauty Products	The Ordinary Niacinamide Serum	1	6.5	6.5	Europe	PayPal
2	10227	2024-08-14	Beauty Products	The Ordinary Caffeine Solution 5% + EGCG	1	6.69999980926514	6.69999980926514	Europe	PayPal
3	10137	2024-05-16	Beauty Products	The Ordinary Hyaluronic Acid Serum	1	6.80000019073486	6.80000019073486	Europe	PayPal
4	10113	2024-04-22	Beauty Products	Biore UV Aqua Rich Watery Essence Sunscreen	1	15	15	Europe	PayPal
5	10167	2024-06-15	Beauty Products	Neutrogena Hydro Boost Water Gel	1	16.9899997711182	16.9899997711182	Europe	PayPal
6	10130	2024-05-09	Books	The Four Agreements by Don Miguel Ruiz	2	8.98999977111816	17.9799995422363	North America	Credit Card
7	10191	2024-07-09	Beauty Products	Glossier Cloud Paint	1	18	18	Europe	PayPal
8	10172	2024-06-20	Books	The Girl with the Dragon Tattoo by Stieg Larsson	2	9.98999977111816	19.9799995422363	North America	Credit Card
9	10232	2024-08-19	Books	Dune by Frank Herbert	2	9.98999977111816	19.9799995422363	North America	Credit Card
10	10070	2024-03-10	Books	The Great Gatsby by F. Scott Fitzgerald	2	10.9899997711182	21.9799995422363	North America	Credit Card

Select \* FROM Data WHERE Transaction\_ID IS NULL OR Date IS NULL OR Product\_Category IS NULL OR Product\_Name IS NULL OR Units\_Sold IS NULL OR Unit\_Price IS NULL OR Total\_Revenue IS NULL OR Region IS NULL OR Payment\_Method IS NULL;

```
SQLQuery1.sql...Shrish (65))*
1 use [ Sales_Trend];
2 select * from Data;
3
4
5 --Data Exploration
6
7 --Count of rows
8 Select count(*) As Total_Rows from Data; --240
9
10 --Sample data
11 Select TOP 10 * FROM Data ORDER BY Total_Revenue;
12
13 --Null values
14 Select * FROM Data WHERE Transaction_ID IS NULL OR Date IS NULL OR Product_Category IS NULL OR Product_Name IS NULL
15 OR Units_Sold IS NULL OR Unit_Price IS NULL OR Total_Revenue IS NULL OR Region IS NULL OR Payment_Method IS NULL;
16
17 --No nulls present in the dataset
18
19
```

100 % No issues found Ln: 14

Transaction_ID	Date	Product_Category	Product_Name	Units_Sold	Unit_Price	Total_Revenue	Region	Payment_Method
----------------	------	------------------	--------------	------------	------------	---------------	--------	----------------

Select DISTINCT Product\_Category FROM Data ORDER BY Product\_Category;

```
SQLQuery1.sql...Shrish (65))*
19
20 --Different product categories
21 Select DISTINCT Product_Category FROM Data ORDER BY Product_Category;
22
23 --Different product names
24 Select DISTINCT Product_Name, Product_Category FROM Data ORDER BY Product_Name
25
26 --Product names present multiple times
27 Select Product_Name, COUNT(Transaction_ID) AS "Number of Items" FROM Data GROU
28
29
30
31 --Data Cleaning
32
33 --Products with price = 0
34 Select * FROM Data WHERE Unit_Price = 0 OR Units_Sold = 0; --No Items
35
36
37
```

100 % No issues found

Product_Category
1 Beauty Products
2 Books
3 Clothing
4 Electronics
5 Home Appliances
6 Sports

Select DISTINCT Product\_Name, Product\_Category FROM Data ORDER BY Product\_Name;

SQLQuery1.sql...Shrish (65))

```

19
20 --Different product categories
21 Select DISTINCT Product_Category FROM Data ORDER BY Product_Category;
22
23 --Different product names
24 Select DISTINCT Product_Name, Product_Category FROM Data ORDER BY Product_Name;

```

100 % No issues found Lnc: 24 Ch: 1 TABS CRLF

Product_Name	Product_Category
1984 by George Orwell	Books
Adidas 3-Stripes Shorts	Clothing
Adidas Essential Track Pants	Clothing
Adidas FIFA World Cup Football	Sports
Adidas Originals Superstar Sneakers	Clothing
Adidas Originals Trefoil Hoodie	Clothing
Adidas Ultraboost Running Shoes	Clothing
Adidas Ultraboost Shoes	Clothing
Amazon Echo Dot (4th Gen)	Electronics
Amazon Echo Show 10	Electronics
Amazon Fire TV Stick 4K	Electronics
Anastasia Beverly Hills Brow Wiz	Beauty Products
Anker PowerCore Portable Charger	Electronics
Anova Precision Cooker	Home Appliances
Anova Precision Oven	Home Appliances
Apple AirPods Max	Electronics
Apple AirPods Pro	Electronics
Apple iPad Air	Electronics
Apple MacBook Air	Electronics
Apple MacBook Pro 16-inch	Electronics
Apple TV 4K	Electronics
Apple Watch Series 8	Electronics
Atomic Habits by James Clear	Books
Babolat Pure Drive Tennis Racket	Sports
Becoming by Michelle Obama	Books
Biore UV Aqua Rich Watery Essenc...	Beauty Products
Blueair Classic 480i	Home Appliances
Bose QuietComfort 35 Headphones	Electronics
Bose QuietComfort 35 II Wireless H...	Electronics
Bose SoundLink Color Bluetooth S...	Sports

Query executed successfully. IDEAPAD5 (16.0 RTM) IDEAPAD5\Shrish (65) Sales\_Trend 00:00:00 234 rows

--Product names present multiple times

Select Product\_Name, COUNT(Transaction\_ID) AS "Number of Items" FROM Data GROUP BY Product\_Name HAVING count(Transaction\_ID) > 1 ORDER BY count(Transaction\_ID) DESC;

SQLQuery1.sql...Shrish (65))

```

19
20 --Different product categories
21 Select DISTINCT Product_Category FROM Data ORDER BY Product_Category;
22
23 --Different product names
24 Select DISTINCT Product_Name, Product_Category FROM Data ORDER BY Product_Name;
25
26 --Product names present multiple times
27 Select Product_Name, COUNT(Transaction_ID) AS "Number of Items" FROM Data GROUP BY Product_Name HAVING count(Transaction_ID) > 1 ORDER BY count(Transaction_ID) DESC;
28
29
30
31 --Data Cleaning
32
33 --Products with price = 0
34 Select * FROM Data WHERE Unit Price = 0 OR Units Sold = 0; --No Items

```

100 % No issues found Lnc: 26 Ch: 1 TABS CRLF

Product_Name	Number of Items
Dune by Frank Herbert	2
Dyson Supersonic Hair Dryer	2
Garmin Forerunner 945	2
Keurig K-Elite Coffee Maker	2
Sony WH-1000XM4 Headphones	2
The Catcher in the Rye by J.D. Salinger	2
The Girl with the Dragon Tattoo by Stieg Larsson	2
The Silent Patient by Alex Michaelides	2

Query executed successfully. IDEAPAD5 (16.0 RTM) IDEAPAD5\Shrish (65) Sales\_Trend 00:00:00

Select \* FROM Data WHERE Unit\_Price = 0 OR Units\_Sold = 0;

The screenshot shows a SQL query editor window titled "SQLQuery1.sql...Shrish (65)". The query is as follows:

```
25
26 --Product names present multiple times
27 Select Product_Name, COUNT(Transaction_ID) AS "Number of Items" FROM Data GROUP BY Product_Name HAVING count(Transaction_ID)
28
29
30
31 --Data Cleaning
32
33 --Products with price = 0
34 Select * FROM Data WHERE Unit_Price = 0 OR Units_Sold = 0; --No Items
35
36
37
38 --Data Analysis
39
```

Below the query editor, the "Results" tab is active, showing a table with the following columns: Transaction\_ID, Date, Product\_Category, Product\_Name, Units\_Sold, Unit\_Price, Total\_Revenue, Region, Payment\_Method. The status bar indicates "No issues found".

Select CAST(SUM(Total\_Revenue) AS DECIMAL(10,2)) AS Revenue from Data

The screenshot shows a SQL query editor window titled "SQLQuery1.sql...Shrish (65)". The query is as follows:

```
36
37
38 --Data Analysis
39
40
41 --1. Calculating the total revenue.
42 Select CAST(SUM(Total_Revenue) AS DECIMAL(10,2)) AS Revenue from Data --80567.85
43
44 --2. Calculating the total revenue grouped by Product Category.
45 Select Product_Category, CAST(SUM(Total_Revenue) AS DECIMAL(10,2)) AS Revenue FROM Data GROUP BY P
46
47 --3. Calculating the regions with the highest sales
48 Select Region, SUM(Units_Sold) AS Units_Sold, CAST(SUM(Total_Revenue) AS DECIMAL(10,2)) AS Revenue
49
50 --4. Calculating the most sold products.
51 Select Product_Category, Product_Name, CAST(Unit_Price AS DECIMAL(10,2)) AS Price, SUM(Units_Sold)
```

Below the query editor, the "Results" tab is active, showing a table with the following columns: Revenue. The status bar indicates "No issues found". The table contains one row with the value 80567.85.

--2. Calculating the total revenue grouped by Product Category.

Select Product\_Category, CAST(SUM(Total\_Revenue) AS DECIMAL(10,2)) AS Revenue FROM Data GROUP BY Product\_Category, Total\_Revenue ORDER BY Total\_Revenue Desc;

The screenshot shows a SQL query window with the following code:

```
--Data Analysis
--1. Calculating the total revenue.
Select CAST(SUM(Total_Revenue) AS DECIMAL(10,2)) AS Revenue from Data --80567.85
--2. Calculating the total revenue grouped by Product Category.
Select Product_Category, CAST(SUM(Total_Revenue) AS DECIMAL(10,2)) AS Revenue FROM Data GROUP BY Product_Category, Total_Revenue ORDER BY Total_Revenue Desc
--3. Calculating the regions with the highest sales
Select Region, SUM(Units_Sold) AS Units_Sold, CAST(SUM(Total_Revenue) AS DECIMAL(10,2)) AS Revenue from Data GROUP BY Region order by Revenue DESC
--4. Calculating the most sold products.
Select Product_Category, Product_Name, CAST(Units_Sold AS DECIMAL(10,2)) AS Units_Sold, SUM(Units_Sold) AS Units_Sold from Data GROUP BY Product_Category, Product_Name
```

The Results pane displays the following data:

	Product_Category	Revenue
1	Electronics	3099.99
2	Home Appliances	2599.98
3	Electronics	2499.99
4	Electronics	2399.00
5	Electronics	1999.98
6	Sports	1895.00
7	Electronics	1599.99
8	Home Appliances	1599.98
9	Electronics	1499.99
10	Electronics	1499.98
11	Electronics	1299.99
12	Electronics	2399.98
13	Electronics	1199.98
14	Home Appliances	1199.98
15	Electronics	2399.94
16	Sports	999.99
17	Electronics	999.98
18	Electronics	899.99
19	Home Appliances	899.99
20	Electronics	1799.94

Query executed successfully.

--3. Calculating the regions with the highest sales

Select Region, SUM(Units\_Sold) AS Units\_Sold, CAST(SUM(Total\_Revenue) AS DECIMAL(10,2)) AS Revenue from Data GROUP BY Region order by Revenue DESC

The screenshot shows the same SQL query window as before, but the Results pane now displays the following data:

	Region	Units_Sold	Revenue
1	North America	180	36844.34
2	Asia	233	22455.45
3	Europe	105	21268.06

Query executed successfully.

--4. Calculating the most sold products.

Select Product\_Category, Product\_Name, CAST(Unit\_Price AS DECIMAL(10,2)) AS Price, SUM(Units\_Sold) AS Units\_Sold from Data GROUP BY Product\_Name, Product\_Category, Unit\_Price, Units\_Sold order by Units\_Sold DESC

The screenshot shows a SQL IDE window with a query titled "SQLQuery1.sql...Shrish (65)". The query is as follows:

```
--1. Calculating the total revenue.
Select CAST(SUM(Total_Revenue) AS DECIMAL(10,2)) AS Revenue from Data --80567.85

--2. Calculating the total revenue grouped by Product Category.
Select Product_Category, CAST(SUM(Total_Revenue) AS DECIMAL(10,2)) AS Revenue FROM Data GROUP BY Product_Category, Total_Revenue ORDER BY Total_Reve

--3. Calculating the regions with the highest sales
Select Region, SUM(Units_Sold) AS Units_Sold, CAST(SUM(Total_Revenue) AS DECIMAL(10,2)) AS Revenue from Data GROUP BY Region order by Revenue DESC

--4. Calculating the most sold products.
Select Product_Category, Product_Name, CAST(Unit_Price AS DECIMAL(10,2)) AS Price, SUM(Units_Sold) AS Units_Sold from Data GROUP BY Product_Name, Pr

--5. Calculating the top 10 products with highest unit price
Select DISTINCT Top 10 Product_Name, Product_Category, CAST(Unit_Price AS DECIMAL(10,2)) AS Price, Units_Sold FROM Data ORDER BY CAST(Unit_Price AS
```

The results table shows the top 10 products by units sold:

Product_Category	Product_Name	Price	Units_Sold
Clothing	Hanes ComfortSoft T-Shirt	9.99	10
Clothing	Gap Essential Crewneck T-Shirt	19.99	6
Clothing	Nike Air Force 1	89.99	6
Sports	Spalding NBA Street Basketball	24.99	6
Sports	Yeti Rambler Tumbler	39.99	6
Sports	Titleist Pro V1 Golf Balls	49.99	5
Clothing	Tommy Hilfiger Polo Shirt	49.99	5
Clothing	Under Armour HeatGear T-Shirt	29.99	5
Sports	Wilson Evolution Basketball	29.99	5
Clothing	Forever 21 Graphic Tee	12.99	5

Query executed successfully. IDEAPADS (16.0 RTM) | IDEAPADS\Shrish (65) | Sales\_Trend | 00:00:00 | 240 rows

--5. Calculating the top 10 products with highest unit price

Select DISTINCT Top 10 Product\_Name, Product\_Category, CAST(Unit\_Price AS DECIMAL(10,2)) AS Price, Units\_Sold FROM Data ORDER BY CAST(Unit\_Price AS DECIMAL(10,2)) DESC;

The screenshot shows a SQL IDE window with a query titled "SQLQuery1.sql...Shrish (65)". The query is as follows:

```
--2. Calculating the total revenue grouped by Product Category.
Select Product_Category, CAST(SUM(Total_Revenue) AS DECIMAL(10,2)) AS Revenue FROM Data GROUP BY Product_Category, Total_Revenue ORDER BY Total_Reve

--3. Calculating the regions with the highest sales
Select Region, SUM(Units_Sold) AS Units_Sold, CAST(SUM(Total_Revenue) AS DECIMAL(10,2)) AS Revenue from Data GROUP BY Region order by Revenue DESC

--4. Calculating the most sold products.
Select Product_Category, Product_Name, CAST(Unit_Price AS DECIMAL(10,2)) AS Price, SUM(Units_Sold) AS Units_Sold from Data GROUP BY Product_Name, Pr

--5. Calculating the top 10 products with highest unit price
Select DISTINCT Top 10 Product_Name, Product_Category, CAST(Unit_Price AS DECIMAL(10,2)) AS Price, Units_Sold FROM Data ORDER BY CAST(Unit_Price AS

--6. Use EXTRACT(MONTH FROM order_date) for month.
Select Date, SUM(Units_Sold) AS Items_Sold from Data Group by Date order by Date
```

The results table shows the top 10 products by unit price:

Product_Name	Product_Category	Price	Units_Sold
Canon EOS R5 Camera	Electronics	3899.99	1
MacBook Pro 16-inch	Electronics	2499.99	1
Apple MacBook Pro 16-inch	Electronics	2399.00	1
Peloton Bike	Sports	1895.00	1
HP Spectre x360 Laptop	Electronics	1599.99	1
Samsung Odyssey G9 Gaming Monitor	Electronics	1499.99	1
LG OLED TV	Home Appliances	1299.99	2
Microsoft Surface Laptop 4	Electronics	1299.99	1
Apple MacBook Air	Electronics	1199.99	1
Samsung QLED 4K TV	Electronics	1199.99	1

Query executed successfully. IDEAPADS (16.0 RTM) | IDEAPADS\Shrish (65) | Sales\_Trend | 00:00:00 | 10 rows

--6. Use EXTRACT(MONTH FROM order\_date) for month.

Select Date, SUM(Units\_Sold) AS Items\_Sold from Data Group by Date order by Date

The screenshot shows a SQL IDE window with a query editor and a results pane. The query editor contains several SQL queries, with the sixth query highlighted: `Select Date, SUM(Units_Sold) AS Items_Sold from Data Group by Date order by Date`. The results pane displays a table with two columns: 'Date' and 'Items\_Sold'. The table contains 20 rows of data, showing dates from 2024-01-01 to 2024-01-20 and corresponding item counts. The status bar at the bottom indicates 'Query executed successfully.' and '240 rows'.

Date	Items_Sold
2024-01-01	2
2024-01-02	1
2024-01-03	3
2024-01-04	4
2024-01-05	1
2024-01-06	5
2024-01-07	1
2024-01-08	2
2024-01-09	6
2024-01-10	2
2024-01-11	1
2024-01-12	3
2024-01-13	2
2024-01-14	1
2024-01-15	2
2024-01-16	3
2024-01-17	1
2024-01-18	4
2024-01-19	2
2024-01-20	1

--7. Calculating Monthly revenue.

Select Datepart(Month, Date) As Month, SUM(Cast(Total\_Revenue as DECIMAL (10,2))) AS Total\_Revenue from Data Group by Datepart(Month, Date) order by Total\_Revenue DESC

The screenshot shows a SQL IDE window with a query editor and a results pane. The query editor contains several SQL queries, with the seventh query highlighted: `Select Datepart(Month, Date) As Month, SUM(Cast(Total_Revenue as DECIMAL (10,2))) AS Total_Revenue from Data Group by Datepart(Month, Date) order by Total_Revenue DESC`. The results pane displays a table with two columns: 'Month' and 'Total\_Revenue'. The table contains 8 rows of data, showing months from 1 to 7 and corresponding total revenue values. The status bar at the bottom indicates 'Query executed successfully.' and '8 rows'.

Month	Total_Revenue
1	14548.32
3	12849.24
4	12451.69
2	10803.37
5	8455.49
6	7384.55
8	7278.11
7	6797.08

--8. Calculating Monthly units sold.

Select Datepart(Month, Date) As Month, SUM(Units\_Sold) AS Total\_Units\_Sold from Data Group by Datepart(Month, Date) order by SUM(Units\_Sold)

The screenshot shows a SQL query in IDEAPADS. The query is as follows:

```
--8. Calculating Monthly units sold.
Select Datepart(Month, Date) As Month, SUM(Units_Sold) AS Total_Units_Sold from Data Group by Datepart(Month, Date) order by SUM(Units_Sold)
```

The results table shows the following data:

Month	Total_Units_Sold
8	52
7	53
2	60
3	61
4	65
5	68
6	77
1	82

The status bar at the bottom indicates "Query executed successfully." and "IDEAPADS (16.0 RTM) | IDEAPADS\Shrish (65) | Sales\_Trend | 00:00:00 | 8 rows".

--9. Calculating Top 2 Months by Monthly revenue.

Select Top 2 Datepart(Month, Date) As Month, SUM(Cast(Total\_Revenue as DECIMAL (10,2))) AS Total\_Revenue from Data Group by Datepart(Month, Date) order by Total\_Revenue DESC

The screenshot shows a SQL query in IDEAPADS. The query is as follows:

```
--9. Calculating Top 2 Months by Monthly revenue.
Select Top 2 Datepart(Month, Date) As Month, SUM(Cast(Total_Revenue as DECIMAL (10,2))) AS Total_Revenue from Data Group by Datepart(Month, Date) order by Total_Revenue DESC
```

The results table shows the following data:

Month	Total_Revenue
1	14548.32
3	12849.24

The status bar at the bottom indicates "Query executed successfully." and "IDEAPADS (16.0 RTM) | IDEAPADS\Shrish (65) | Sales\_Trend | 00:00:00 | 2 rows".



--10. Calculating Top 3 Days by revenue.

Select Top 3 Datepart(WEEKDAY, Date) As WEEKDAY, SUM(Cast(Total\_Revenue as DECIMAL (10,2))) AS Total\_Revenue from Data Group by Datepart(WEEKDAY, Date) order by Total\_Revenue DESC

The screenshot shows a SQL query window with the following text:

```
--7. Calculating Monthly revenue.
Select Datepart(Month, Date) As Month, SUM(Cast(Total_Revenue as DECIMAL (10,2))) AS Total_Revenue from Data Group by Datepart(Month, Date) order by Total_Revenue DESC

--8. Calculating Monthly units sold.
Select Datepart(Month, Date) As Month, SUM(Units_Sold) AS Total_Units_Sold from Data Group by Datepart(Month, Date) order by SUM(Units_Sold)

--9. Calculating Top 2 Months by Monthly revenue.
Select Top 2 Datepart(Month, Date) As Month, SUM(Cast(Total_Revenue as DECIMAL (10,2))) AS Total_Revenue from Data Group by Datepart(Month, Date) order by SUM(Cast(Total_Revenue as DECIMAL (10,2)))

--10. Calculating Top 3 Days by revenue.
Select Top 3 Datepart(WEEKDAY, Date) As WEEKDAY, SUM(Cast(Total_Revenue as DECIMAL (10,2))) AS Total_Revenue from Data Group by Datepart(WEEKDAY, Date) order by SUM(Cast(Total_Revenue as DECIMAL (10,2)))

--11. Calculating the Weekdays having highest units sold.
Select Datepart(Weekday, Date) As Weekday, SUM(Units_Sold) AS Total_Units_Sold from Data Group by Datepart(Weekday, Date) order by SUM(Units_Sold)
```

The results pane shows the following data:

	WEEKDAY	Total_Revenue
1	3	12518.34
2	6	12918.74
3	2	12253.35

The status bar at the bottom indicates: Query executed successfully. IDEAPADS (16.0 RTM) IDEAPADS\Shrish (65) Sales\_Trend 00:00:00 3 rows

--11. Calculating the Weekdays having highest units sold.

Select Datepart(Weekday, Date) As Weekday, SUM(Units\_Sold) AS Total\_Units\_Sold from Data Group by Datepart(Weekday, Date) order by SUM(Units\_Sold)

The screenshot shows a SQL query window with the following text:

```
--9. Calculating Top 2 Months by Monthly revenue.
Select Top 2 Datepart(Month, Date) As Month, SUM(Cast(Total_Revenue as DECIMAL (10,2))) AS Total_Revenue from Data Group by Datepart(Month, Date) order by SUM(Cast(Total_Revenue as DECIMAL (10,2)))

--10. Calculating Top 3 Days by revenue.
Select Top 3 Datepart(WEEKDAY, Date) As WEEKDAY, SUM(Cast(Total_Revenue as DECIMAL (10,2))) AS Total_Revenue from Data Group by Datepart(WEEKDAY, Date) order by SUM(Cast(Total_Revenue as DECIMAL (10,2)))

--11. Calculating the Weekdays having highest units sold.
Select Datepart(Weekday, Date) As Weekday, SUM(Units_Sold) AS Total_Units_Sold from Data Group by Datepart(Weekday, Date) order by SUM(Units_Sold)
```

The results pane shows the following data:

	Weekday	Total_Units_Sold
1	4	65
2	5	70
3	2	71
4	6	74
5	3	76
6	1	78
7	7	84

The status bar at the bottom indicates: Query executed successfully. IDEAPADS (16.0 RTM) IDEAPADS\Shrish (65) Sales\_Trend 00:00:00 7 rows